

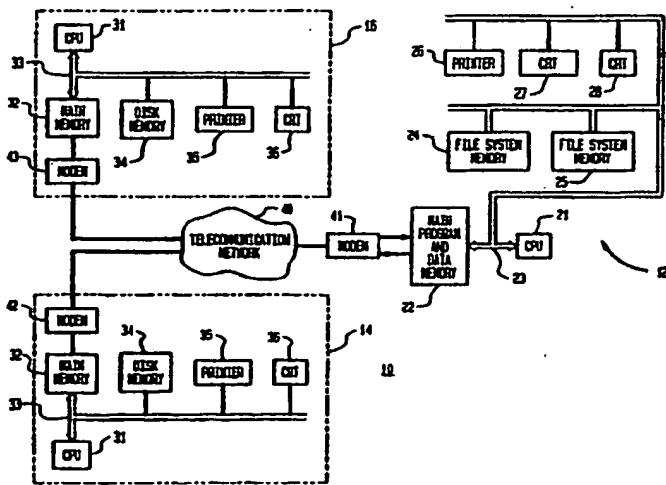
PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06F 13/00, 9/45, 12/14	A1	(11) International Publication Number: WO 92/22033
		(43) International Publication Date: 10 December 1992 (10.12.92)
<p>(21) International Application Number: PCT/US92/00935</p> <p>(22) International Filing Date: 6 February 1992 (06.02.92)</p> <p>(30) Priority data: 705,188 24 May 1991 (24.05.91) US</p> <p>(71) Applicant: BELL COMMUNICATIONS RESEARCH, INC. (US/US); 290 West Mount Pleasant Avenue, Livingston, NJ 07039-2729 (US).</p> <p>(72) Inventor: BORENSTEIN, Nathaniel, Solomon ; 25 Washington Avenue, Morristown, NJ 07960 (US).</p> <p>(74) Agents: WINTER, Richard, C.; PCT International, Inc., Post Office Box 573, New Vernon, NJ 07976 (US) et al.</p>		<p>(81) Designated States: AT (European patent), BE (European patent), CA, CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, LU (European patent), MC (European patent), NL (European patent), SE (European patent).</p> <p>Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p>

(54) Title: ACTIVE MESSAGING SYSTEM



(57) Abstract

A computer system (12) for receiving active messages comprises a central processing unit (21) and peripheral devices (24, 25, 26, 27, 28) connected to the central processing unit, such as a file memory system (24, 25) for storing a file system of a recipient of the active message. The computer system includes an interpreter (70) for interpreting instructions contained in an active message to convert the instructions into machine instructions to be executed by said central processing unit. To prevent an active message from causing a deprivation of resources, the instruction set which is interpreted by the interpreter is limited. The interpreter interprets a subset of instructions of a general purpose computer language which does not relate to the inputting and outputting of data to and from peripheral devices. The interpreter interprets a set of input/output instructions which can only achieve limited access to the file system and which controls the amount of data which is transmitted to and from peripheral devices.

- 1 -

ACTIVE MESSAGING SYSTEM

Field of the Invention

5 Conventional electronic mail is in the form of a passive textual message that is created by an originator, transmitted to a recipient and read by the recipient. Any further actions must be initiated by the recipient. In contrast, an active message is a program which is created
10 by an originator, transmitted to a recipient, and executed in the recipient's computer environment. The present invention relates to an active messaging system and method which enables a recipient to enjoy the benefits of active messaging while reducing the security threats posed by
15 active messaging.

Background of the Invention

In the growing field of office systems, no system is considered complete unless it has an electronic mail facility. Currently available electronic mail systems
20 allow passive messages to be composed and sent to other

- 2 -

users in the system or network. A passive message, i.e., a piece of text conveying some information, is created by a sender and shipped by electronic mail to one or more recipients. A passive message can only convey information, 5 it cannot collect information. The task of a passive message is complete when it is read by a recipient.

In contrast to a passive message, an active message is a program that is run. It carries on a dialogue with a recipient, asking questions and collecting answers. The 10 sender does not need to depend upon the recipient creating another passive message in response to some question, as this may be done by the active message itself. Conventional electronic mail messaging is static. The original sender specifies a list of recipients and the 15 message is sent to exactly the destinations on that list. In contrast, because an active message is a program, in the course of being run on behalf of one recipient, the active message may route itself to other recipients.

Active messaging can provide numerous benefits to 20 users. One important use of active messaging is to implement a communal memory in which a user sends messages to potential "experts" to elicit answers to questions. If one potential expert does not know the answer, the active message will try to elicit the identities of additional 25 potential experts and cause itself to be transmitted to these additional potential experts. When an expert who knows the answer is found, the active message causes the

- 3 -

answer to be transmitted back to the original sender of the question.

Another important application of active messaging is to automate the most routine and annoying aspects of scheduling meetings or other current events. Currently, 5 meetings are scheduled as a result of an extensive exchange of passive messages, e.g., "Can you meet on Tuesday at 3:00 PM?" "Tuesdays are bad for me, how about Wednesday morning?" "Wednesday morning is no good, how about the afternoon?" Such exchanges can be automated using active 10 messaging. For example, an active message can make the rounds of all the participants who are supposed to attend a meeting. From each one, the active message collects one or more plausible dates and times for the meeting. The active 15 message then sends a copy of itself, augmented with the collected data, to the next participant. Eventually, possibly after several iterations, the active message will find a date and time when all the participants can meet. The active message will then inform all the participants of 20 this date and time.

Active messaging may also be utilized for paperwork automation. Many organizations have large numbers of routine processes that are handled by massive amounts of paperwork. A significant portion of such paperwork has to 25 be routed through several organizational centers in sequence for approvals or other kinds of intermediate actions. Using active messaging, much of this processing

- 4 -

could be automated. An active message can be received at one organizational center for asking a few questions to collect certain information. The active message then sends itself off to the next link in the organizational chain.

5 For example, expense vouchers, purchase orders, and insurance claims can be processed in this manner.

A still further application of active messaging is the collection of information for opinion surveys.

While active messaging has many potential benefits, it is also evident that if active message programs are written in a sufficiently powerful language, enormous harm can result. For example, someone could send a message which deletes all the files of the recipient. Even more insidiously, one could create a mail based virus that could bring any computer network to its knees, simply by mailing out two copies of itself every time it is received by a recipient. Other dangers posed by active messaging includ the deprivation of file resources as by overwriting existing files or by filling up the file system of a recipient with data, the deprivation of CPU time by transmitting an active message which takes a long time to execute, and the deprivation of I/O resources such as a printer by transmitting an active message which causes a large amount of material to be printed. In many cases, such dangers overwhelm any possible benefit that might be found in utilizing activ m ssages. Therefore, in order for active messaging to become a useful r ality, a system

- 5 -

must be devised wherein the utilization of active messages cannot result in unacceptable damage to the active message recipients.

A variety of active messaging systems have previously
5 been proposed. However, none of the prior systems have satisfactorily resolved the security problems resulting from active messaging. One active messaging system known as "R2D2" has no provisions for security (see, e.g., John Vittal, "Active Message Processing: Messages As
10 Messengers", in Computer Message Systems, R.P. Uhlig, editor, North-Holland Publishing Company, 1981). Another active messaging system is known as "imail" (see, e.g., John Hogg et al, "An Active Mail System", Proceedings of SIGMOD '84, Boston 1984). The "imail" system gains security from
15 a rather crucial restriction. Active messages can only be exchanged among users of a single machine using specialized software.

Specialized active messaging functionality, i.e., a mail system that implements one or a few special cases of
20 active messaging, such as return-receipt mail, requests for votes, and other specific types of active messages have also been developed. One example of such a specialized active messaging system is the "Andrew" system (see e.g., Boerenstein, et al, "Architectural Issues in the Andrew
25 Message Systems"; E. Stefferud et al, "Message Handling Systems and Distributed Applications"; and Bor nst in et al, "Power, Ease of Use and Cooperative Work in a Practical

- 6 -

Multimedia Message System", Int. J. Man-Machine Studies (1991) 34, pp. 229-259). This type of system avoids problems in security by defining a very specialized syntax for a single kind of non-extensible active message.

5 Therefore the capability of doing great harm, such as by deleting all of the recipients' files or by creating a virus, is not present in such systems. A more general system based on the specialized Andrew system is known as "Ness" (see e.g., W. Fred Hansen, "Enhancing Documents With

10 Embedded Programs: How Ness Extends Insets in the Andrew Took Kit", Proceedings of IEEE Computer Society, 1990, International Conference in Computer Language, New Orleans, 1990). "Ness" is probably the most powerful of the prior art active messaging systems, but it offers only a token solution to the security problem. When a recipient reads a message with a Ness program inside, the recipient is asked if he/she trusts the author of the program and is given the opportunity to read the code before executing it. Ness programs have the capability of doing great harm such as by

15 deleting the recipient files. Thus if a recipient wrongly indicates trust, great harm can occur.

20

Finally, there is the Strudel system (see, e.g., Alan Shepherd et al, "Strudel - An Extensible Electronic Conversion Took Kit", Proceedings of CSCW '90, October 1990). The Strudel system allows arbitrary LISP expressions to be sent through the mail so that Strudel is

- 7 -

very p w rful. However, because of this, Strudel also has the power to cause great harm.

In view of the foregoing, it is an object of the present invention to provide a general active messaging system which provides active message recipients with an acceptable level of security.

Summary of the Invention

In accordance with an illustrative embodiment of the present invention, a computer system for receiving and processing electronic mail in the form of active messages comprises a central processing unit and one or more peripheral devices connected to the central processing unit. The peripheral devices include for example a file memory system for storing a file system of a recipient of an active message. The peripheral devices also include input/output devices for interfacing the computer system with the outside world such as printers, display terminals, and interfaces with external networks.

The computer system for processing active messages includes a mail reader for reading electronic mail received at the computer system via a communications channel. If the electronic mail contains only a passive message, the mail reader causes the passive message to be displayed for the recipient, for example, on the recipient's display terminal.

- 8 -

The mail read r may also indicate that the received electronic mail comprises an active message in the form of a program which is made up of one or more instructions to be executed by the central processing unit. In this case an interpreter is utilized to interpret the instructions contained in the active message so that the instructions may be executed by the central processing unit. The interpreter converts the instructions into machine language instructions for this purpose. Depending on the language in which the instructions are written, the instructions might be converted into machine language instructions by a compiler rather than an interpreter.

As indicated above, active messages pose a security threat to their recipients because they have the capability of depriving the recipient of computer resources. For example, an active message may access the file system of the recipient to delete or write over certain files, or simply write an extremely large amount of data into the file system. Similarly, an active message may cause the deprivation of CPU time by transmitting a program which takes a large amount of time to execute. Alternatively, the active message may deprive a recipient of an input/output resource such as a printer by causing a large amount of material to be printed.

To eliminate such a deprivation of resources in accordance with the present invention, the interpreter can only interpret a limited set of instructions. If an active

- 9 -

message contains an instruction which is not in the instruction set of the interpreter, the instruction will not be executed by the central processing unit. In particular, the interpreter of the present invention can 5 interpret the subset of a general purpose computer language such as LISP, which subset does not include instructions relating to the transmitting of data to or from the peripheral devices such as the file system memory and I/O devices. The interpreter of the present invention can also 10 interpret a set of instructions to access the peripheral devices in a manner which does not cause a deprivation of resources. Thus, the interpreter can interpret instructions for accessing only a predetermined limited portion of the file system of the recipient of the active 15 message. Within this predetermined portion of the recipient's file system, files can be read, new files can be created (subject to limitations on their size and number), but no files can be deleted or overwritten. The interpreter also places limitation on the number of bytes 20 which can be transferred to a peripheral device by an instruction of an active message. In addition, there may also be a limit on the amount of execution time required for the instructions of an active message.

When constructed in the foregoing manner, an active 25 messaging system provides a high level of generality for its users so that it may be utilized in a wide variety of

- 10 -

applications, while protecting recipients against serious resource deprivation problems.

Brief Description of the Drawing

5 FIG 1 schematically illustrates a network in which active messages may be transmitted among users.

FIG 2 schematically illustrates the programs that are executed to transmit and receive active messages.

10 FIG 3 schematically illustrates an interpreter that may be utilized to interpret the instructions comprising active messages, in accordance with an illustrative embodiment of the present invention.

Detailed Description of the Invention

15 FIG 1 illustrates a network 10 in which active messaging may be utilized by various users. The network 10 comprises the main frame system 12 and the workstations 14 and 16.

20 The main frame system 12 comprises a CPU 21 and a main program and data memory 22. The memory 22 stores programs being executed by the CPU 21 and is utilized in connection with the execution of these programs. The mainframe computer system 12 includes a number of peripheral devices which are connected to the CPU 21 and main memory 22 via
25 the local area network 23. The peripheral devices include the memory systems 24 and 25. The memory systems 24 and 25 are illustratively implemented by magnetic disks and the ir

- 11 -

associated drives. The memory systems 24 and 25 store the file systems of the various users of the mainframe computer system 12. Also connected to the local area network 23 are a plurality of I/O devices such as the printer 26 and the 5 display terminals 27 and 28. Instead of the local area network 23, the elements comprising the main frame system 12 may be interconnected by a bus system or other interconnection medium.

In addition to comprising the mainframe computer system 10 12, the network 10 of FIG 1 includes the workstations 14 and 16. Each workstation includes a CPU 31, a main memory 32, and a bus system 33. Connected to the bus system 33 are a disk memory 34, a printer 35, and a display terminal 36.

15 Users of the computer system 12 can send electronic mail to and receive electronic mail from the workstations 14 and 16 via the telecommunications network 40 which illustratively is the public switched telephone network. For this purpose, the computer system 12 and the 20 workstations 14 and 16 include the modems 41, 42, and 43, respectively, for interfacing with the telecommunications network 40. It is also possible for one user of the computer system 12 to send mail to another user of the computer system 12 via the local area network 23 or bus or 25 other interconnection medium. In addition, instead of utilizing the public switch d telephone network, mail may

- 12 -

be transmitted between the systems 12, 14, and 16 via another network such as Ethernet or the Arpa network.

Consider the example where the user of the workstation 14 wishes to send an electronic mail message to a recipient 5 who is a user of the computer system 12. As shown in FIG 2, an active or passive message may be generated at the workstation 14 by the user by typing on the keyboard associated with display terminal 36. Alternatively, an active message may be generated automatically at the 10 workstation 14 by an active mail generator 50. In any of these cases, the message to be sent is transmitted to the electronic mailer 52. The mailer 52 then transmits a piece 15 of electronic mail containing the message via modem 42 and the telecommunications network 40 to the computer system 12. In the case of the Internet electronic mail system (see David H. Crocker "Standard for the Format of Arpa Internet Text Messages," Network Information Center RFC #822 1982; Marvin A. Sirbu "Content-Type Header Field for Internet Messages", Network Information Center RFC #1049, 20 1988), active messages are tagged with an active message header field.

The message arrives at the computer system 12 via the modem 41 (see FIG 1) and main memory 22 and is stored under the control of the CPU 21 in a file system of the intended 25 recipient, which file system is maintained in the memory 24 or 25.

- 13 -

The programs which are executed when the recipient at the computer system 12 receives his/her mail are illustrated in FIG 2. The first program which is executed by the recipient who wishes to read his/her mail is the 5 mail reader 60 which for example is a conventional UNIX mail reader. In the instance of a passive message, the mail reader 60 (which in order to be executed is resident in the main memory 22 of FIG 1), fetches the recipient's messages from the file system memory 24 or 25 (see FIG 1) 10 and stores the messages in the main memory 22. As shown in FIG 2, the message is transmitted by the mail reader 60 to a display program 62 so that the message is displayed, for example, on the recipient display terminal which may be the display terminal 27 or 28.

15 If a message is an active message as indicated by an active message header, the active message is transmitted by the mail reader 60 to the active message interpreter 70. The active message interpreter 70 interprets the instructions contained in the active message by converting 20 the instructions into machine language instructions which are executed by the CPU 21. The active message instructions may include instructions for processing data and instructions for performing input/output operations.

To prevent the recipient from experiencing a 25 deprivation of resources from the execution of an active message, the active message interpreter can only execute a certain limited set of instructions. Thus, the active

- 14 -

messaging system and method of the present invention work best when the transmitter of an active message composes the active message from the instruction set of the interpreter 70 of FIG 2. If an active message contains instructions outside the instruction set of the interpreter, such instructions will not be executed and execution of the active message will be halted.

The set of instructions which can be executed by the interpreter 70 of FIG 2 may be described as follows. Illustratively, the instruction set of the interpreter 70 includes the instructions of a general purpose computer language such as LISP except for instructions relating to the input/output of data to and from various peripheral devices such as the memories 24, 25, the printer 26, and the display terminals 27 and 28. The instruction set of the interpreter 70 also includes a set of instructions for accessing the peripheral devices in a manner which does not lead the recipient to experience a deprivation of computer resources. Thus, the interpreter 70 can only execute instructions for accessing a special limited subdirectory of a recipient's file system. Files elsewhere in a recipient's file system simply do not exist as far as the interpreter 70 is concerned. Within the limited file system that is accessible to active messages, any files can be read. Files can also be created subject to limitations on size and number. However, files cannot be deleted or overwritten. A further restriction on file names involves

- 15 -

the use of symbolic links. On certain systems such as UNIX, users can create symbolic links that make files outside the special subdirectory for active messaging accessible through the special subdirectory. In some 5 embodiments of the invention, the interpreter restricts symbolic links so that file names which follow symbolic links beyond a single step are not interpreted. That is, a user can create symbolic links to specific files outside the special active message subdirectory but entire 10 directory structures cannot be made so readable.

In addition, the instruction set of the interpreter 70 of FIG 2 is limited so that there is a limit on the total number of bytes that may be written into the active message directory area of a recipient and a limit on the number of 15 bytes that may be sent to a printer or other peripheral device to produce output.

A complete instruction set for an active messaging interpreter is based on the general purpose and widely available language known as LISP. In particular, to arrive 20 at the instruction set starting with the instruction set of LISP, the subset of instructions not related to data input/output are chosen and a set of instructions related to input/output of data which will not cause a deprivation of resources are also included.

- 16 -

It is a significant advantage of the active messaging system and method of the present invention that the instruction set of the interpreter is derived from the instruction set of a general purpose computer language such as LISP. This makes the active messaging system highly portable over a wide variety of user interface platforms, operating systems, hardware environments and file systems.

FIG 3 is a flowchart which schematically illustrates the operation of the interpreter 70 of FIG 2. Each instruction of an active message to be executed is fetched from a main memory (e.g. 22) associated with the CPU (e.g. 21) which executes the instruction (step 100 of FIG 3). (As indicated previously, the instructions comprising an active message are transferred from a file system to the main memory to be executed). A test is then made to determine if the fetched instruction is safe (step 110 of FIG 3). This test is implemented by determining if the fetched instruction is in the instruction set of the interpreter. If the instruction is not in the instruction set, execution is halted (step 120). If the instruction is in the instruction set, the instruction is executed (step 130).

The step 110 may be implemented by means of a table lookup, i.e., to carry out the step 110, for each instruction in an active message, a table lookup is performed to see if the instruction is in the interpreter's instruction set. Note that an instruction may be outside

- 17 -

the instruction set because an operator is excluded or because an operand is excluded such as in the case of an instruction which tries to output more than a predetermined number of bytes to a printer.

5 The execution step 130 usually involves converting the active message instructions to machine language instructions which are then executed by the CPU (e.g. 21). In the case where the instruction set is derived from the LISP language, the execution step 130 may be implemented through use of a LISP engine, for example, ELI (Embedded 10 LISP Interpreter) which is available for example from IBM.

15 After an instruction is executed (step 130), a test is performed to determine if there are any instructions remaining in the message (step 140). If so, control is returned to the step 100 and the next instruction is fetched from the main memory.

20 In some embodiments of the invention, it may be desirable to give a recipient of a message more control over whether or not certain instructions are or are not executed, i.e., for some instructions the recipient may be asked to state whether or not the instruction should be executed. The portion of the flowchart relating to this feature is designated by 200 in FIG 3.

25 In this case, an instruction which is not indicated as safe by the step 110, is subjected to a test (step 210) to determine if the instruction is potentially safe. This test may be implemented through use of a second table

- 18 -

lookup which contains a list of the potentially safe instructions. A potentially safe instruction is an instruction which may be safe or unsafe depending on the specific operators and operands and depending on the specific circumstances of the message recipient. If an instruction fails the test 110 and the test 210, execution is terminated (step 120). If an instruction fails the test 110 but the test 210 indicates that the instruction is potentially safe, the recipient is asked if he/she wants the instruction executed (step 220). If the recipient desires the instruction to be executed, then control passes to step 130. If the recipient does not want the instruction executed, control passes to the step 120.

An example of an instruction which is potentially safe
15 is an instruction for sending mail. Therefore, in accordance with an illustrative embodiment of the present invention, whenever an active message running in a recipient's computer environment tries to send mail, the recipient is told of this attempt and given an opportunity to inspect the mail and decide whether the mail should be sent. Without this restriction, it would be easy to create viruses, chain letters and other undesirable phenomena
20 using active messaging.

In addition to the foregoing restrictions implemented
by means of an interpreter, a limitation may be placed on
the amount of CPU time which can be taken by an active
message. While in some embodiments of the invention this

- 19 -

limitation can be incorporated in the interpreter, when the UNIX operating system is utilized, it may be easier to implement this limitation directly through use of operating system facilities.

5 In short, a method and system for carrying out active messaging is disclosed. The system and method enable a recipient to enjoy the many benefits of active messaging while reducing the security risks of active messaging. Finally, the above-described embodiments of the inventions
10 are intended to be illustrative only. Numerous alternative embodiments may be devised by those skilled in the art without departing from the spirit and scope of the following claims.

- 20 -

CLAIMS

1. A computer system for receiving and processing information containing an active message made up of
5 instructions to be executed, said computer system comprising:

a central processing unit and one or more peripheral devices connected to said central processing unit,

10 means for receiving said information via a communications channel at said computer system, and converting means for determining if each instruction in said active message belongs to a predetermined set of instructions, for converting the instructions in said active message belonging to said set into a form suitable
15 for execution by said central processing unit, and for inhibiting the execution of said instructions in said active message not belonging to said set,

20 said predetermined set of instructions comprising an instruction set of a general purpose computer language but excluding instructions relating to the transfer of data to or from said one or more peripheral devices beyond predetermined limits.

25 2. The computer system of claim 1 wherein said converting means comprises an interpreter.

3. The computer system of claim 2 wherein said converting means comprises a compiler.

- 21 -

4. The system of claim 1 wherein one of said peripheral devices is a file system memory connected to said central processing unit for storing a file system of a recipient of said active message, and wherein said predetermined limits define a predetermined portion of said file system of said recipient, so that said predetermined instruction set includes instructions for accessing said predetermined file portion of said recipient and excludes instructions for accessing the remainder to the file system of said recipient.

5. The system of claim 4 wherein said predetermined set of instructions include instructions for reading any file in said predetermined portion of said file system of said recipient, instructions for creating new files of a predetermined size and number in said predetermined portion of said file system of said recipient, but excludes instructions for deleting files or overwriting existing files.

15. The system of claim 5 wherein said predetermined limits include a limit on the number of bytes which may be written into said predetermined portion of said file system of said recipient.

20. The system of claim 1 wherein said system includes means for limiting the execution of the instructions contained in said active message to a predetermined amount of central processing time.

- 22 -

8. The system of claim 1 wherein said predetermined set of instructions include an instruction for sending electronic mail.

5 9. The system of claim 8 wherein instructions for sending electronic mail is an instruction for sending electronic mail containing an active message.

10. The system of claim 4 wherein said predetermined set of instructions include an instruction for accessing said file system of said recipient containing a one-step symbolic link.

11. A computer system for receiving and processing information containing an active message made up of instructions to be executed, said computer system comprising:

15 a central processing unit and one or more peripheral devices connected to said central processing unit, means for receiving said information via a communications channel at said computer system, and an interpreter for determining if each instruction in said active message belongs to a predetermined set of instructions, for converting the instructions in said active message belonging to said set into a form suitable for execution by said central processing unit, and for inhibiting the execution of said instructions in said 20 active message not belonging to said set,

25

- 23 -

said predetermined set of instructions comprising an instruction set of general purpose computer language, but excluding instructions relating to the transfer of data to or from said peripheral devices beyond predetermined limits.

5

12. The system of claim 11 wherein said interpreter includes means for enabling a recipient of an active message to control whether selected instructions contained in the active message are executed.

10 13. A method for processing information containing an active message made up of instructions to be executed comprising the steps of

receiving said active message at a computer system via a communications channel,

15 translating said instructions in said active message which belong to a predetermined set of instructions into a form suitable for execution by a central processing unit forming part of said computer system, and

inhibiting the execution of instructions contained 20 in said active message not belonging to said set,

25 said predetermined set of instructions comprising an instruction set of a general purpose computer language but excluding instructions relating to the transfer of data to or from one more peripheral devices connected to said central processing unit beyond predetermined limits.

- 24 -

14. The method of claim 13 wherein said method includes the step of utilizing a table lookup to determine if an instruction in said active message belongs to said predetermined set of instructions.

1/3

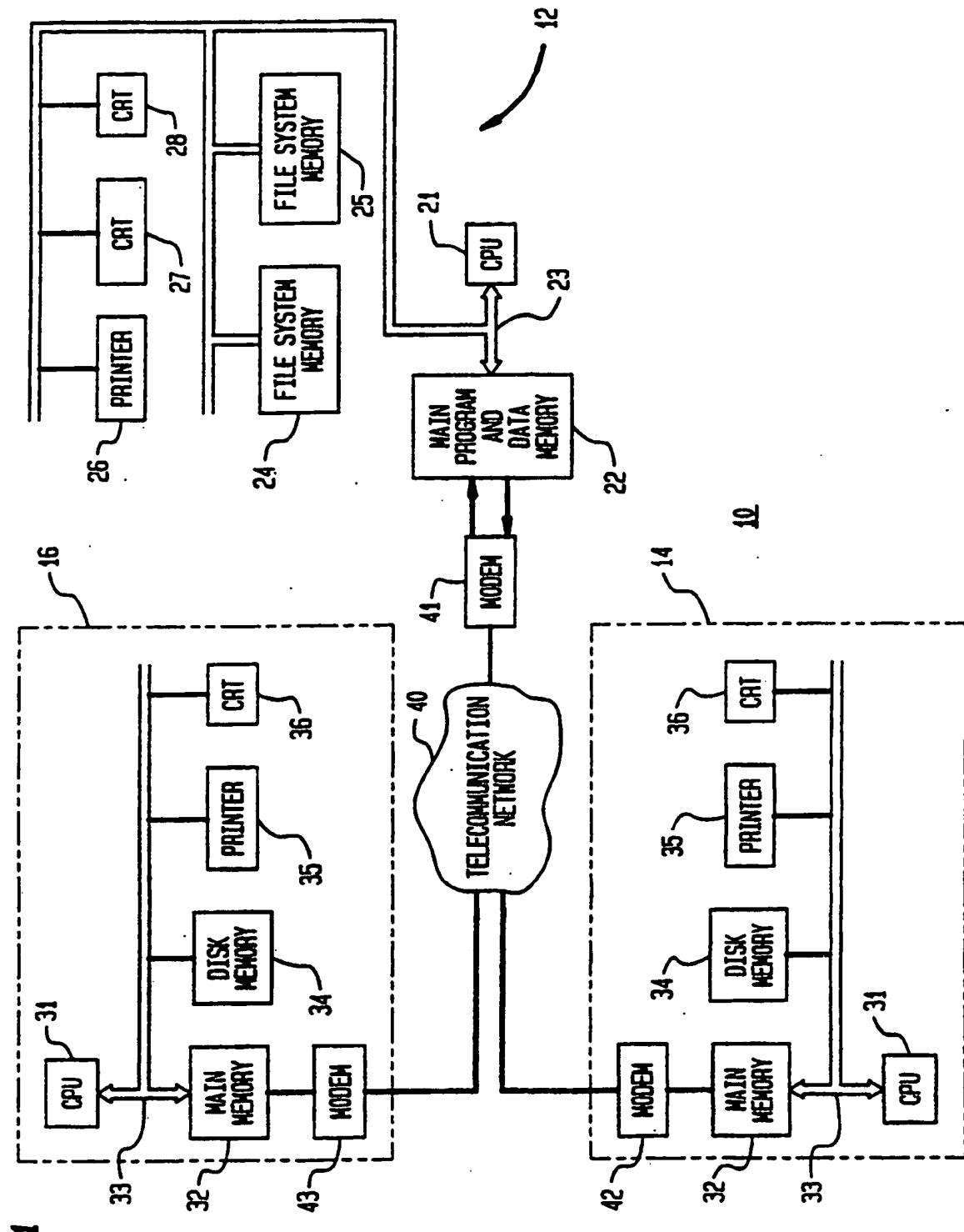


FIG. 2

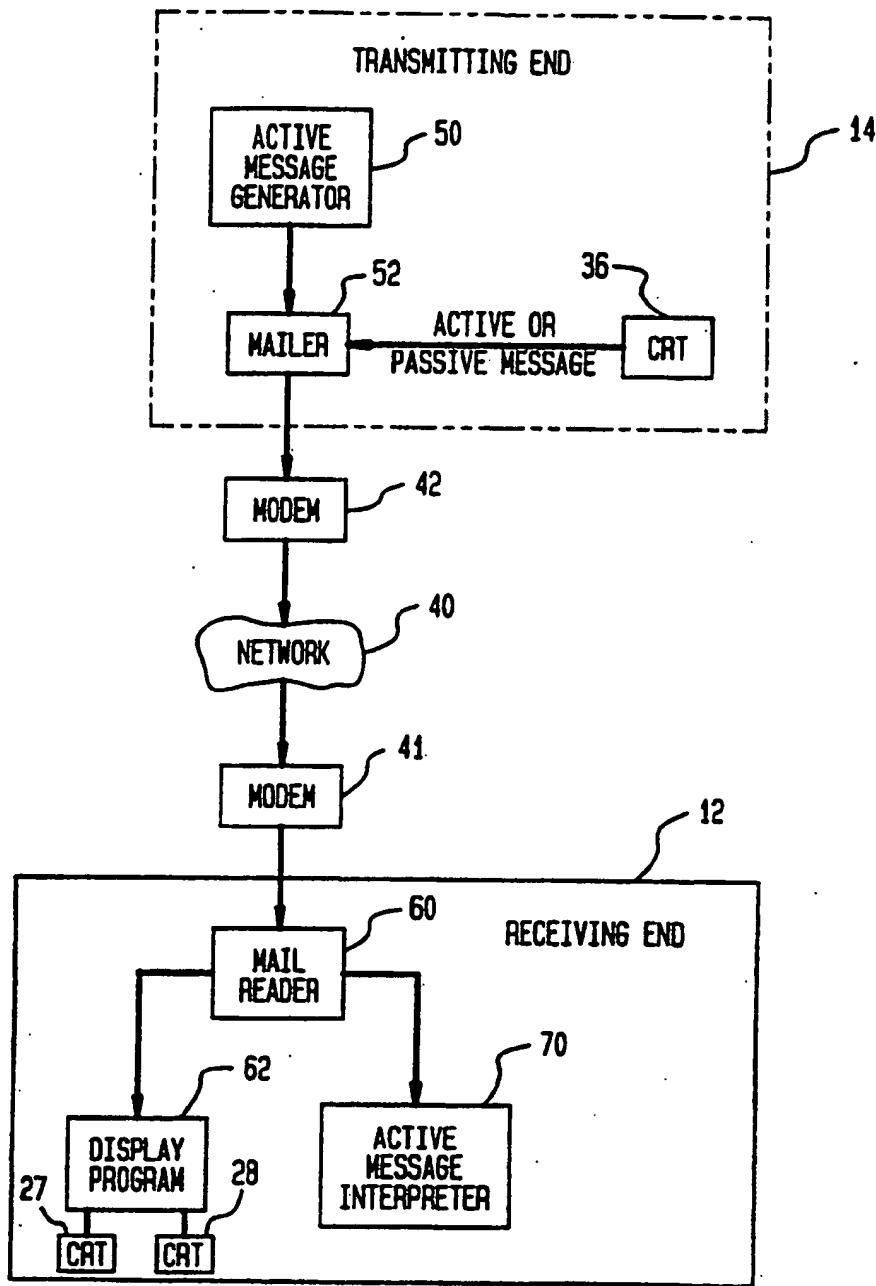


FIG. 3

